

État du support de l'espace fine insécable dans l'écosystème du logiciel libre

Nicolas DELVAUX

Dernière mise à jour: 15 août 2011

Remerciements

Je remercie chaleureusement les divers contributeurs de la liste de diffusion *traduc.org* pour leur aide.

Je tiens à remercier plus particulièrement William GATHOYE pour les nombreuses corrections et améliorations qu'il m'a permis d'apporter ainsi que pour avoir réalisé tous les tests sous Mac OS.

Avant propos

Les sources de ce document (rédigé en L^AT_EX) sont disponibles à l'adresse suivante : <http://malizor.org/public/fines/>

N'hésitez pas à m'envoyer vos remarques et corrections (éventuellement sous formes de patches) à l'adresse suivante : contact@nicolas-delvaux.org

Attention, les discussions sur l'adoption (ou non) de l'espace fine insécable ne doivent pas se dérouler en privé avec moi, mais plutôt sur les listes de diffusions des divers groupes de traducteurs.

Table des matières

Remerciements	i
Avant propos	ii
Introduction	1
I Mode graphique	3
1 Protocole de test	3
2 Résultats	4
2.1 Gnome/Applications GTK+	4
2.1.1 Applications communes	4
2.1.2 Documentation	4
2.1.3 GConf	5
2.1.4 Windows	5
2.1.5 Bilan	6
2.2 KDE/Applications Qt	7
2.2.1 Applications communes	7
2.2.2 Windows	7
2.2.3 Bilan	8
2.3 Plateforme Mozilla/XUL	9
2.3.1 GNU/Linux	9
2.3.2 Windows	9
2.3.3 Bilan	10
2.4 wxWidgets	11
2.4.1 GNU/Linux	11
2.4.2 Windows	11
2.4.3 Bilan	12
2.5 Tcl/Tk	13
2.5.1 GNU/Linux	13
2.5.2 Windows	13
2.5.3 Bilan	14
2.6 Le cas Mac OS	15
II Mode texte	17
3 Protocole de test	17
4 Résultats	18
4.1 Gnome-terminal	18
4.2 Terminator	18
4.3 Xfce4-terminal	18

4.4	Konsole	18
4.5	Xterm	18
4.6	Mrxvt	18
4.7	TTY	19
4.8	Bilan	19
III Rendu de pages web		20
5	Protocole de test	20
6	Résultats	21
6.0.1	WebKitGTK+	21
6.0.2	KHTML/QtWebKit	21
6.0.3	Firefox	22
6.0.4	Safari	23
6.0.5	Internet Explorer	23
6.0.6	Bilan	25
Conclusion		26

Introduction

Ce document présente les résultats des divers tests qui ont été effectués afin de connaître l'étendue du support actuel de l'espace fine insécable dans le monde du logiciel libre.

Il est bien entendu illusoire de prétendre être exhaustif, aussi ai-je tenté de sélectionner les applications les plus représentatives possible sur les systèmes d'exploitation en ma possession¹.

Je vous encourage à compléter ce document avec vos propres expérimentations. Il serait notamment souhaitable que quelques tests soient réalisés avec un système BSD (même si j'imagine, peut-être à tort, que les résultats seraient les mêmes que sous GNU/Linux).

Qu'est-ce qu'une espace fine insécable ?

Dans l'intégralité de ce document, une « espace fine insécable » représente le caractère Unicode « U+202F NARROW NO-BREAK SPACE » apparu en 1999 avec Unicode 3.0.

Son emploi est recommandé en français devant les caractères « ;?! et » et après le caractère « , », et cela par des institutions telles que que l'Imprimerie nationale (voir le livre *Lexique des règles typographiques en usage à l'Imprimerie nationale*) ou encore l'université Paris Diderot (voir cette page : <http://www.eila.univ-paris-diderot.fr/sysadmin/gestion-docs/typo-francaise#ponctuation>).

Mais son emploi ne doit pas uniquement être motivé par la « rigueur académique », car l'espace fine insécable apporte tout simplement une meilleure cohérence, une meilleure « harmonie » dans les chaînes de caractères où elle est employée en lieu et place de l'espace insécable classique voire d'aucune espace du tout.²

Sur un système GNU/Linux avec l'agencement « France/France Autre »³, le raccourci clavier correspondant est « Alt Gr + v ».

Il est également possible (bien que moins pratique) de saisir une espace fine insécable sous Windows.

Pour se faire, il faut faire un réglage dans le registre : allez dans « HKEY_Current_User/Control Panel/Input Method » grâce à « REGEDIT » et ajoutez (si elle n'existe pas déjà) une nouvelle clef de type « String » ayant pour nom « EnableHexNumpad » et pour valeur « 1 ».

Ensuite, après un redémarrage, vous pourrez saisir une espace fine insécable dans un programme tel que « Notepad » en maintenant la touche « Alt » puis en tapant « +202f » et enfin en relâchant « Alt ».⁴

1. William Gathoye s'est occupé des tests sous Mac OS

2. C'est certes aussi une question de goût, mais cette opinion semble être partagée par de nombreux « connaisseurs ».

3. Agencement francophone par défaut sur de nombreuses distributions

4. Il existe d'autres méthodes parfois plus simples, mais celles-ci sont uniquement propres à certains logiciels (voir cette page : http://www.fileformat.info/tip/microsoft/enter_unicode.htm). La méthode décrite ici a l'avantage d'être théoriquement universelle.

De quoi dépend son support ?

Le support correct de l'espace fine insécable dépend en fait d'une seule chose : le moteur de rendu.

On pense souvent que les polices ont un rôle à jouer. C'est en pratique vrai dans certains cas, mais cela ne devrait pas être nécessaire avec tout « bon » moteur de rendu : en rencontrant une fine, celui-ci devrait automatiquement afficher une espace insécable en divisant la taille de la police courante par 2 voir 3, ou encore en réutilisant le glyphe correspondant à l'espace fine sécable « U+2009 THIN SPACE » (et en le considérant comme étant insécable).¹

Le cas échéant, il est de plus beaucoup plus facile de corriger les moteurs de rendu que les polices (celles-ci étant beaucoup plus nombreuses, souvent propriétaires et peu maintenues).

Je ne me suis donc pas ici attaché au test des polices.²

1. À ma connaissance, il s'agit de la solution qui a été retenue pour le moteur de rendu intégré à Windows 7. Je ne sais pas ce qu'il en est pour les autres moteurs de rendu.

2. Un test rapide, peut-être non significatif, dans mon éditeur de texte m'a tout de même permis de constater que l'écrasante majorité des polices (et la totalité des polices usuelles, incluant celles de Microsoft) la supporte. (testé dans gedit sous Ubuntu 10.04)

Première partie

Mode graphique

1 Protocole de test

Dans le monde graphique, le moteur de rendu de textes utilisé par une application peut être soit intégré au système d'exploitation soit intégré à la « bibliothèque graphique » (*toolkit*) utilisée.

Nous allons donc passer en revue les principaux *toolkits* sur divers systèmes d'exploitation.

Les tests sur GNU/Linux ont été réalisés sur un système Ubuntu 10.04 à jour, aucun des réglages par défaut susceptible d'affecter les tests n'a été modifié, que ce soit au niveau des polices (hormis des augmentations ponctuelles de la taille pour des raisons de lisibilité), des moteurs de rendu et autres.

Même si à ma connaissance, dans ce domaine, Ubuntu n'impose pas de réglages exotiques par défaut par rapport à *upstream*, il pourrait être intéressant de réitérer tout ou partie de ces tests sur d'autres distributions/systèmes d'exploitation afin de corroborer les résultats.

Les environnements Windows testés sont les suivants :

- Windows 7 à jour (virtualisé grâce à Virtualbox¹)
- Windows XP à jour (*dual-boot*)
- Wine² (version 1.2) installé sur le système Ubuntu 10.04 décrit précédemment

Les tests sur Mac OS ont été réalisés sur une version 10.6 à jour virtualisée grâce à VMware Workstation 7.

1. <http://www.virtualbox.org/>

2. <http://www.winehq.org/>

2 Résultats

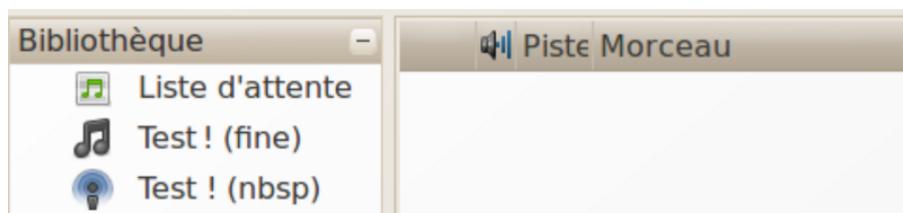
2.1 Gnome/Applications GTK+

D'après mes recherches, GTK utilise Pango¹ pour tout ce qui a trait au rendu de textes.

C'est donc vraisemblablement lui qui sera indirectement testé dans cette section.

2.1.1 Applications communes

Voici l'exemple d'une application commune utilisant GTK+ : Rhythmbox (version 0.12.8)



Pas de problème.

2.1.2 Documentation

Je souhaitais effectuer un autre test sur une application dont les traductions sont « compilées » en fichiers `.xml` en lieu et place des traditionnels `.mo`.

Voici le lecteur de documentation de Gnome : Yelp² (version 2.30.0)

Glossaire

Test ! (fine) Test ! (nbsp)

Une applet est une petite application
Contrôleur de volume. Chaque applet
du clavier.

Pas de problème.

1. <http://www.pango.org/>
2. <http://live.gnome.org/Yelp>

2.1.3 GConf

Les schémas des clefs GConf sont encore à part au sein de la plateforme Gnome. Test réalisé sous gconf-editor¹ (version 2.30.0) sur une clef du logiciel Evince² :



On remarque que la fine affichée ici est identique à l'espace insécable.

La police n'étant visiblement pas à chasse fixe, il y a donc un bug de rendu ici. Cependant, comme gconf-editor va être remplacé par dconf-editor à l'occasion de la sortie de Gnome 3, on peut espérer que ce problème sera réglé au passage par l'utilisation de technologies plus récentes.

De plus, bien qu'étant défectueux, le rendu de la fine dans gconf-editor n'introduit pas de problème vis à vis de l'utilisateur par rapport à la situation actuelle.

Le relatif échec de ce test n'est donc, à priori, pas un obstacle à l'adoption de la fine au sein des applications GTK et de la plateforme Gnome.

2.1.4 Windows

GTK (et donc Pango) étant multiplateforme, je me suis basé sur le logiciel Pidgin³ (version 2.7.3) pour réaliser quelques tests dans différents environnements Windows.

Voici tout d'abord le rendu obtenu sous Ubuntu :



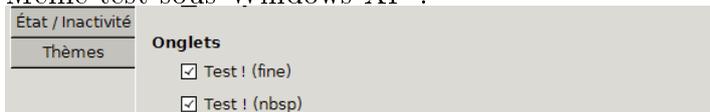
La différence n'est pas suffisamment marquée, mais sinon pas de problème.

Même test sous Windows 7 :



Idem.

Même test sous Windows XP :



Idem.

1. <http://projects.gnome.org/gconf/>
2. <http://projects.gnome.org/evince/>
3. <http://pidgin.im/>

Même test sous Wine :



Idem.¹

2.1.5 Bilan

Hormis un relatif échec en ce qui concerne GConf, c'est un sans faute. Il semblerait que les applications basées sur GTK+ (et donc sur Pango) soient d'ores et déjà prêtes pour adopter l'espace fine insécable sous GNU/Linux et Windows (voir la partie « Le cas Mac OS »). Ceci est particulièrement intéressant pour la plateforme Gnome qui pourrait ainsi dès maintenant fournir une meilleure « expérience typographique » à ses nombreux utilisateurs.

1. Le rendu des widgets et des polices est certes assez effroyable, mais la fine est correctement affichée.

2.2 KDE/Applications Qt

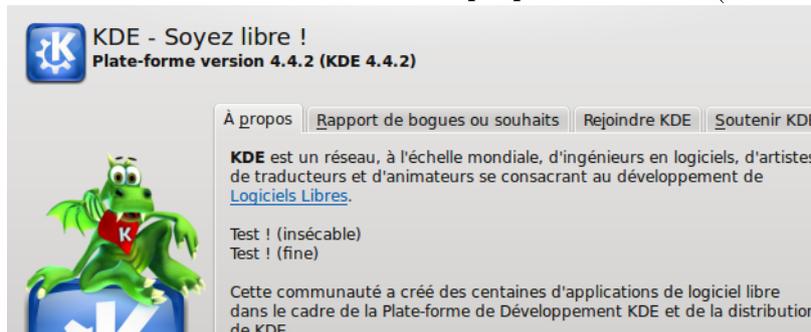
Je me dois de reconnaître ici ma relative méconnaissance de cet environnement.

Je n'ai ici réalisé que peu de tests ; si d'autres s'avèreraient pertinents, contactez moi !¹

D'après mes recherches, il semble que le rendu de textes soit ici directement intégré à Qt².

2.2.1 Applications communes

Test réalisé sur la fenêtre « À propos de KDE » (version 4.4.2)



On constate que l'espace fine n'est pas prise en compte.

Ce bug de rendu n'introduit donc pas d'effet néfaste par rapport à la situation actuelle.

Par acquis de conscience, j'ai réalisé un autre test sur l'application Konsole³ (version 2.4.2)



On constate ici un bug de rendu beaucoup plus grave : l'espace fine est tout simplement ignorée !

J'ignore cependant le pourquoi de cette différence de rendu avec le test précédent, il faudrait que des connaisseurs se penchent sur la question.

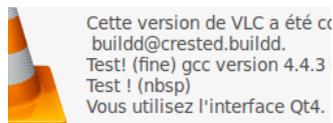
2.2.2 Windows

Je me suis ici basé sur la célèbre application VLC⁴ (version 1.1.3).

VLC est multiplateforme et son interface est basée sur Qt. C'est donc le cobaye idéal !

-
1. Ou mieux, réalisez les et envoyez moi les résultats !;-)
 2. <http://qt.nokia.com/>
 3. <http://konsole.kde.org/>
 4. <http://www.videolan.org/vlc/>

Voici tout d'abord le rendu obtenu sous Ubuntu :



```
Cette version de VLC a été cc
buildd@crested.buildd.
Test! (fine) gcc version 4.4.3
Test ! (nbsp)
Vous utilisez l'interface Qt4.
```

Même bug que lors du test de Konsole.

Même test sous Windows 7 :

```
Test! (fine) gcc version 4.4.4
Test ! (nbsp)
Vous utilisez l'interface Qt4.
```

Idem.

Même test sous Windows XP :

```
Test! (fine) gcc version 4.4.4
Test ! (nbsp)
Vous utilisez l'interface Qt4.
```

Idem.

Même test sous Wine :

```
Test! (fine) gcc version 4.4.4
Test ! (nbsp)
Vous utilisez l'interface Qt4.
```

Idem.

2.2.3 Bilan

Ce bilan est malheureusement négatif pour la plateforme Qt et KDE. Rien de désespéré cependant, il s'agit vraisemblablement d'un seul et unique bug dans le moteur de rendu intégré à Qt. Le corriger devrait donc régler tous les problèmes observés ici (tout du moins, espérons-le).

J'ai ouvert un rapport de bug à cette adresse : <http://bugreports.qt.nokia.com/browse/QTBUG-13280>.

Toute aide pour sa résolution est la bienvenue, vous pouvez également aller « voter » pour ce bug (il y a un lien sur la gauche de la page web) afin d'attirer l'attention des développeurs.

En attendant, il semble malheureusement exclu d'utiliser les espaces fines insécables sur cette plateforme.

2.3 Plateforme Mozilla/XUL

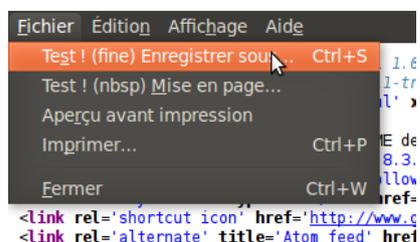
Les produits et technologies Mozilla¹ sont très présents de nos jours, il est donc naturel que leur compatibilité avec l'espace fine insécable soit également testée.

J'ai ici basé mes tests sur le très célèbre navigateur Firefox², dans sa version 3.6.8. Je n'ai pas testé Thunderbird³, mais les deux logiciels étant basé sur un même « cœur », les résultats ne devraient pas être différents.

Je n'ai de plus pas ici testé le rendu des pages web contenant des espaces fines insécables car celui-ci est indépendant de la possibilité de traduction des applications en elles mêmes.

Référez vous à la partie « Rendu de pages web » pour plus d'informations sur le sujet.

2.3.1 GNU/Linux

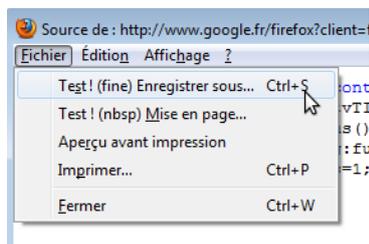


La différence n'est pas suffisamment marquée, mais sinon pas de problème.

On pouvait toute fois s'y attendre car, d'après Wikipedia⁴, les versions GNU/Linux de Firefox et de Thunderbird utilisent Pango pour le rendu de textes (cf. la section « Gnome/Applications GTK+ »).

2.3.2 Windows

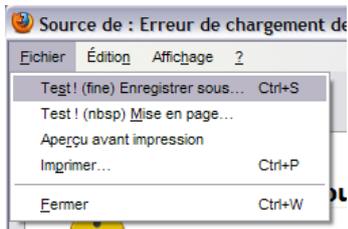
Même test sous Windows 7 :



Idem.

1. <http://www.mozilla.org/>
2. <http://www.firefox.com/>
3. <http://www.getthunderbird.com/>
4. <http://en.wikipedia.org/wiki/Pango#Utilization>

Même test sous Windows XP :



Idem.

Même test sous Wine :



Wine fait ici dans l'originalité puisqu'il nous affiche une fine plus large que l'espace insécable classique !

2.3.3 Bilan

Un bilan très positif car, hormis un petit ratage sous Wine, c'est un sans faute. Les produits Mozilla étant multiplateforme il n'y a, à priori, pas à s'inquiéter de ce problème.

En faisant abstraction du « cas Mac OS » (cf. la partie correspondante), il semble donc que les traductions de ces logiciels puissent adopter l'espace fine insécable dès maintenant.

2.4 wxWidgets

wxWidgets¹ est une autre bibliothèque graphique libre et multiplateforme. Elle est par exemple utilisée dans des applications telles que FileZilla² ou encore Poedit³.

Les tests ci-dessous ont été réalisés avec FileZilla dans sa version 3.3.1 pour Ubuntu et dans sa version 3.3.4.1 pour les autres systèmes.

2.4.1 GNU/Linux



La différence n'est pas suffisamment marquée, mais sinon pas de problème.

2.4.2 Windows

Même test sous Windows 7 :



Idem.

Même test sous Windows XP :



Gros problème de rendu.

-
1. <http://www.wxwidgets.org/>
 2. <http://filezilla-project.org/>
 3. <http://www.poedit.net/>

Même test sous Wine :



Idem.

2.4.3 Bilan

Ce bilan est mitigé, il est probable que le bug de rendu observé sur Windows XP et Wine soit dû au moteur de rendu intégré au système.

Il est cependant rassurant de voir que tout fonctionne dans un environnement « moderne », cela va donc dans le bon sens.

L'utilisation de l'espace fine insécable dans les logiciels utilisant wxWidgets ne semble donc pas recommandable pour l'instant, sauf pour les applications ne fonctionnant que sur GNU/Linux.

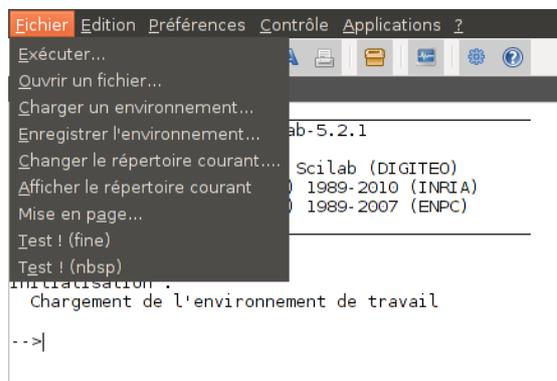
2.5 Tcl/Tk

Même si Tcl/Tk n'est utilisé que très marginalement par rapport aux géants GTK et Qt, il me semblait tout de même intéressant (ne serait-ce que pour donner une vague impression d'exhaustivité à ce rapport ;-)) de le tester. Tcl/Tk est de plus multiplateforme.

J'ai ici basé mes tests sur le logiciel scientifique de calcul numérique Scilab¹ (version 5.2.1).

Je n'ai pas trouvé d'informations qui suggèreraient que Scilab utilise un autre système de rendu que celui intégré par défaut à Tcl/Tk (l'idéal serait tout de même de le confirmer).

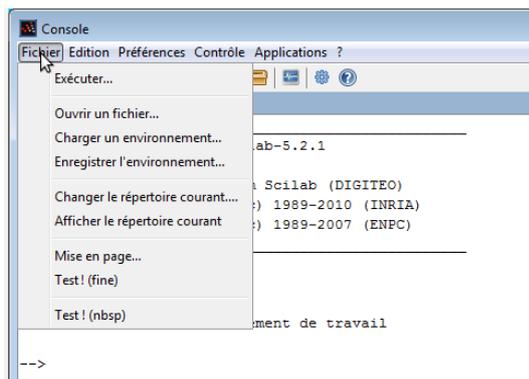
2.5.1 GNU/Linux



La différence n'est pas suffisamment marquée, mais sinon pas de problème.

2.5.2 Windows

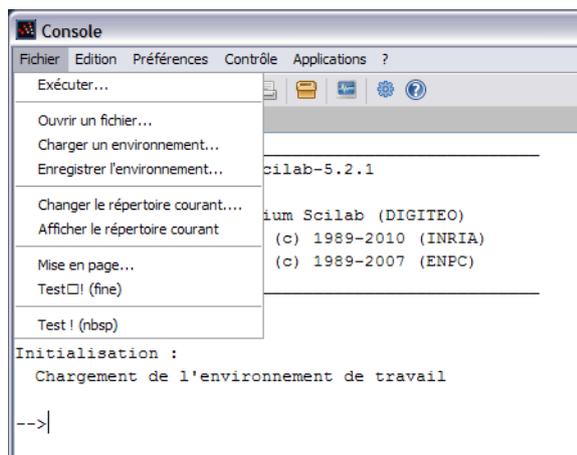
Même test sous Windows 7 :



Pas de problème.

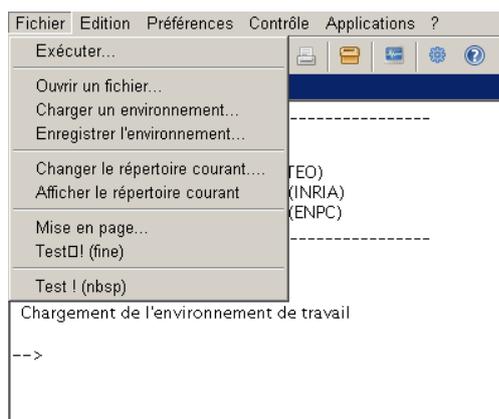
1. <http://www.scilab.org/>

Même test sous Windows XP :



Gros problème de rendu.

Même test sous Wine :



Idem.

2.5.3 Bilan

Les résultats sont les mêmes que ceux observés avec wxWidgets. Le bilan est donc mitigé.¹

Il est cependant rassurant de voir que tout fonctionne dans un environnement « moderne », cela va donc dans le bon sens.

L'utilisation de l'espace fine insécable dans les logiciels utilisant Tcl/Tk ne semble donc pas recommandable pour l'instant, sauf pour les applications ne fonctionnant que sur GNU/Linux.

1. J'ai également voulu tester aMSN, mais j'ai découvert que celui-ci n'est même pas compatible UTF-8! De plus, sa traduction actuelle ne met aucune espace entre les signes de ponctuations (ce qui n'est en fait pas étonnant car, d'après mes tests, même les espaces insécables classiques ne s'affichent pas correctement).

2.6 Le cas Mac OS

D'après mes recherches, le rendu de textes des applications Mac OS est effectué par Quartz¹ (cela reste tout de même à confirmer).

Quartz fait partie des fondations de Mac OS, il est donc utilisé dans toutes les API de plus haut niveau telles que « Cocoa » ou « Carbon ».

Sans plus attendre, voici un test réalisé dans la barre de menu globale de Mac OSX 10.6 (vraisemblablement programmée avec Cocoa) :

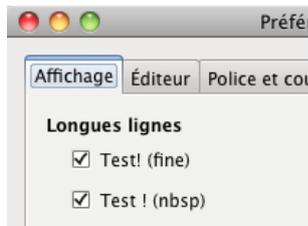


Le résultat est négatif, rien n'est affiché...

Mais il y a plus problématique encore : il semble que, dans leur version Mac, la plupart des bibliothèques graphiques tierces reposent sur Cocoa/Carbon et donc sur Quartz. Les versions Mac des applications GTK, wxWidgets et autres héritent donc également de ce bug, même si leurs versions GNU/Linux et Windows fonctionnent correctement !

Voici quelques captures d'écran pour tenter d'étayer mon propos.

Commençons tout d'abord par Gedit² (version 2.30.2) qui est une application GTK+³.



Voici maintenant Firefox (XUL) :



1. http://en.wikipedia.org/wiki/Quartz_%28graphics_layer%29
2. <http://projects.gnome.org/gedit/>
3. Je confirme au passage qu'il supporte correctement l'espace fine insécable sur les autres autres systèmes d'exploitation

Terminons par FileZilla (wxWidgets) :



Ces 3 exemples semblent confirmer mon hypothèse.

Il serait possible de modifier l'ensemble des *toolkits* pour qu'ils contournent ce problème sous Mac OS, mais la tâche serait considérable. Corriger le problème à la source semble donc être la solution la plus viable.

J'ai rapporté le bug sur le site des développeurs Apple¹, espérons qu'il sera pris en compte (vu que nous n'avons pas ici affaire à un logiciel libre, il nous est impossible de corriger le problème nous même...)

En attendant, ce bug reste important.

Cependant, Mac OS n'est pas non plus une plateforme de prédilection pour de nombreux projet. Sachant que ce bug se contente de ne rien afficher à la place d'une espace fine insécable (il aurait été plus problématique s'il affichait un autre glyphe à la place), peut-être n'est-il pas non plus nécessaire de le considérer comme bloquant pour des programmes/projets qui fonctionnent par ailleurs très bien sur les autres systèmes d'exploitation (Gnome, par exemple).

La décision appartient aux équipes de traduction compétentes.

1. Je ne peux malheureusement pas vous vous fournir de lien car il semble que les rapports de bugs ne soient pas publiques chez Apple.

Deuxième partie

Mode texte

3 Protocole de test

Ces tests ont été réalisés sur un système Ubuntu 10.04 à jour, aucun des réglages par défaut susceptible d'affecter les tests n'a été modifié, que ce soit au niveau des polices, des moteurs de rendu et autres.

Même si à ma connaissance, dans ce domaine, Ubuntu n'impose pas de réglages exotiques par défaut par rapport à *upstream*, il pourrait être intéressant de réitérer tout ou partie de ces tests sur d'autres distributions/systèmes d'exploitation.

Les tests suivants ont été effectués sur la célèbre application « wget » (version 1.12), en supposant que les applications du même type réagiraient de manière identique dans le même environnement.

J'ai décidé de modifier la chaîne suivante du fichier `.po` correspondant :¹

```
#: src/main.c:908 src/main.c:977 src/main.c:1099
#, c-format
msgid "Try '%s --help' for more options.\n"
msgstr "Essayez « %s --help » pour plus d'informations."
```

Pour afficher cette chaîne, il suffit de taper « wget » dans un terminal. Voici maintenant la nouvelle chaîne qui va servir lors des tests :

```
"Essayez « %s --help » pour plus d'informations. (2 nbsp)\n"
"Essayez « wget --help » pour plus d'informations. (2 fines)\n"
```

Pour des raisons techniques, la deuxième ligne devait contenir « wget » directement à la place de « %s ». Cela n'a aucun impact sur les résultats du test.

J'ai essayé de sélectionner un échantillon représentatif des différentes consoles disponibles.

Notez que les consoles utilisent traditionnellement des polices à chasse fixe, il est donc normal qu'aucune différence ne soit visible entre les espaces fines insécables et les espaces insécables « classiques ».

1. Notez que la traduction actuelle accole des espaces simples (sécables) aux guillemets.

4 Résultats

4.1 Gnome-terminal

Terminal par défaut sous Gnome (version 2.29.6)

```
Essayez « wget --help » pour plus d'informations. (2 nbsp)
Essayez « wget --help » pour plus d'informations. (2 fines)
```

Pas de problème.

4.2 Terminator

Un autre terminal pour Gnome (version 0.93)

```
Essayez « wget --help » pour plus d'informations. (2 nbsp)
Essayez « wget --help » pour plus d'informations. (2 fines)
```

Pas de problème.

4.3 Xfce4-terminal

Terminal par défaut sous Xfce (version 0.4.3)

```
Essayez « wget --help » pour plus d'informations. (2 nbsp)
Essayez « wget --help » pour plus d'informations. (2 fines)
```

Pas de problème.

4.4 Konsole

Terminal par défaut sous KDE (version 2.4.2)

```
Essayez « wget --help » pour plus d'informations. (2 nbsp)
Essayez « wget --help » pour plus d'informations. (2 fines)
```

Les espaces fines insécables sont simplement ignorées.

4.5 Xterm

Un célèbre terminal pour X (version 256)

```
Essayez « wget --help » pour plus d'informations. (2 nbsp)
Essayez « wget --help » pour plus d'informations. (2 fines)
```

Pas de problème.

4.6 Mrxvt

Un autre terminal pour X (version 0.5.4)

```
Essayez « wget --help » pour plus d'informations. (2 nbsp)
Essayez « wget --help » pour plus d'informations. (2 fines)
```

Sans commentaire...

(vous noterez que même les espaces insécables « classiques » ne sont pas supportées)

4.7 TTY

Dois-je vraiment les présenter ?

```
Essayez « wget --help » pour plus d'informations. (2 nbsp)
Essayez «♦wget --help♦» pour plus d'informations. (2 fines)
```

Étrangement, les fines sont remplacées par des espèces de petits losanges...

Le problème a été identifié comme venant de KBD, un ensemble d'utilitaires de gestion du texte et des polices proche du noyau.

J'ai moi même proposé une correction à ce problème et celle-ci a été acceptée en amont : <http://git.altlinux.org/people/legion/packages/kbd.git?p=kbd.git;a=commitdiff;h=50f674d1775bc75f799c583b887c3329088ff620>

Ce problème ne sera donc plus qu'un mauvais souvenir lorsque KBD aura été mis à jour sur la plupart des systèmes.

4.8 Bilan

Sur les 7 tests effectués, 4 ont parfaitement fonctionné et 3 ont révélé des problèmes d'affichage.

Parmi les 3 échecs, on distingue :

- 1 cas de non-affichage/non-prise en compte de la fine (Konsole), il s'agit certainement d'une autre manifestation du bug de Qt
- 1 cas d'échec total (Mrxvt) où même les espaces insécables classiques ne sont pas affichés correctement
- 1 cas de substitution de la fine par autre chose (TTY)

Ce dernier est certes le plus dérangeant, mais il est maintenant corrigé en amont. Aucun obstacle technique majeur ne semble donc plus s'opposer à la généralisation de l'emploi de l'espace fine insécable dans le monde de la ligne de commande.

Certes, la version corrigée de KBD n'a pas encore (à la date d'aujourd'hui) atteint les distributions GNU/Linux usuelles, mais il est de toute manière probable que celui-ci sera mis à jour d'ici à ce que les traductions des utilitaires en ligne de commande le soit également.

Troisième partie

Rendu de pages web

Le rendu de pages web peut tout d'abord paraître étranger à la traductions de logiciels libres.

Il est vrai que ce domaine à une importance moindre par rapport aux 2 précédents, mais la plupart des *toolkits* incluent maintenant des *widgets* permettant d'intégrer une page web (qu'elle soit présente en local ou hébergée à distance) dans une application.

Certaines applications proposent également une « interface web » pour être contrôlées à distance.

5 Protocole de test

Pour réaliser ces tests, je me suis basé sur la page suivante :

<http://malizor.org/public/fines/fines.html>

J'ai fait afficher cette page sur divers navigateurs. Je me suis également servi du site <http://browsershots.org/>, mais je ne sais pas si ses résultats sont fiables (je ne sais pas si les systèmes sont à jour, s'ils ont été modifiés, si l'intégration dans le système pour prendre les instantanés peut avoir une influence...)

6 Résultats

J'ai ici essayé de sélectionner les navigateurs les plus pertinents. Vous trouverez le reste¹ des images obtenues grâce à <http://browsershots.org/> à l'adresse suivante : <http://malizor.org/public/fines/images/browsershots>

6.0.1 WebKitGTK+

Une bibliothèque de plus en plus utilisée pour intégrer du contenu web dans les applications GTK+.

J'ai ici choisi de tester Midori² (version 0.2.2) car ce navigateur sans fioritures utilise cette fameuse bibliothèque.
(testé sous Ubuntu 10.04)



Pas de problème.³

6.0.2 KHTML/QtWebKit

Il semble que WebKit soit doucement en train de remplacer son vénérable ancêtre KHTML au sein de la plateforme KDE et de Qt.

J'ai tout d'abord testé Konqueror⁴ (version 4.4.2) qui est basé sur KHTML :



1. À l'exception des tests que j'avais déjà effectués, des pages blanches et autre bugs venant de l'outil de test en lui même

2. http://www.twotoasts.de/index.php?/pages/midori_summary.html

3. Pour information, j'ai obtenu le même résultat avec Epiphany, le navigateur par défaut de Gnome.

4. <http://konqueror.kde.org/>

J'ai ensuite testé Rekonq¹ (version 0.4.0) qui est basé sur sur la bibliothèque qt-webkit :



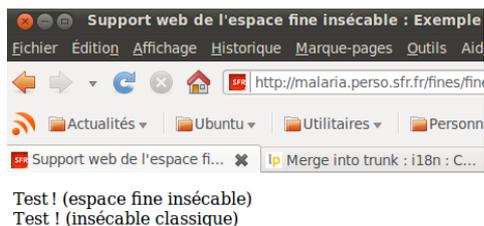
Dans ces deux cas (sur Ubuntu 10.04), le résultat obtenu ne tient pas compte de l'espace fine insécable.

Cela s'explique certainement par le même bug de rendu observé sur Qt, car ces navigateurs en utilisent probablement le moteur de rendu de textes.

6.0.3 Firefox

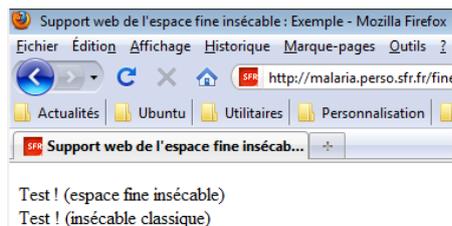
Test de la version 3.6.8.

Sur Ubuntu 10.04 :



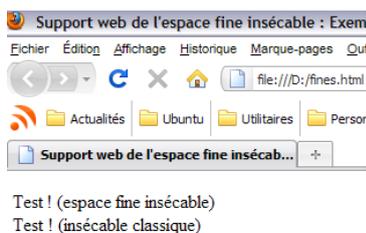
Pas de problème.

Sous Windows 7 :



Pas de problème.

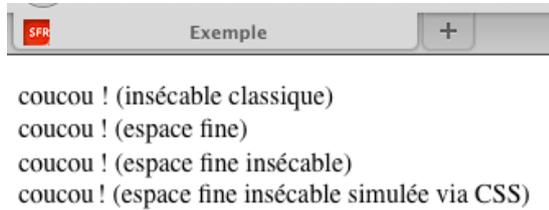
Sous Windows XP :



1. <http://rekonq.sourceforge.net/>

Pas de problème.¹

Sous Mac OSX :



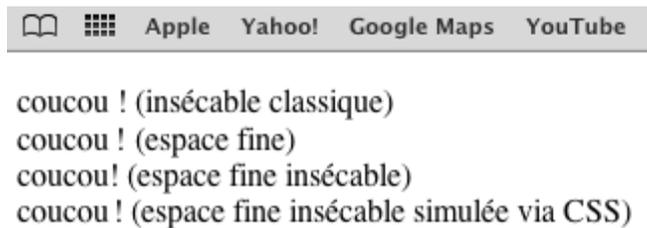
Pas de problème non plus !

Firefox semble donc parfaitement au point vis à vis de l'espace fine insécable.

6.0.4 Safari

Le célèbre navigateur d'Apple (version 5.0.1).

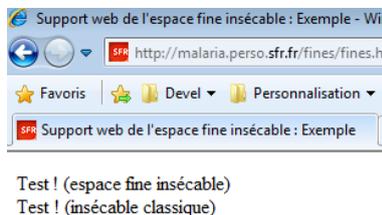
Sous Mac OSX :



Comme on pouvait s'y attendre, Safari ne se comporte pas mieux que les autres applications Mac « natives ».

6.0.5 Internet Explorer

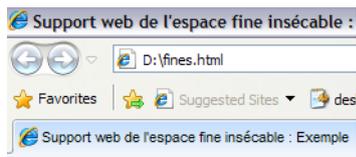
Je terminerais ce bref tour d'horizon par le tristement célèbre navigateur de Microsoft. Je n'ai pu tester que la version 8, cependant nous verrons que cela est suffisant. Sous Windows 7 :



Pas de problème.

Sous Windows XP :

1. Pour des raisons techniques, j'ai dû ouvrir le fichier html en local. Cela ne change normalement rien au test.



Test□! (espace fine insécable)
Test ! (insécable classique)

Problème de rendu.¹

Cela confirme (si c'était encore nécessaire) que le rendu dépend du moteur de rendu de textes.

Internet Explorer étant intégré à Windows, celui-ci en utilise le moteur de rendu. Le moteur de Windows 7 supporte l'espace fine insécable contrairement à celui intégré à Windows XP.

On peut donc vraisemblablement supposer qu'aucune des versions de Internet Explorer sous Windows XP n'affiche l'espace fine insécable correctement.²

1. Pour des raisons techniques, j'ai dû ouvrir le fichier html en local. Cela ne change normalement rien au test.

2. Je ne sais pas contre pas ce qu'il en est pour Windows Vista.

6.0.6 Bilan

Bien que, à mon sens, le support de l'espace fine insécable dans les principaux navigateurs ne soit pas « bloquant » pour son adoption au sein des traductions des logiciels libres, il reste tout de même rassurant de voir que tout fonctionne dans les environnements « modernes ». On peut donc être optimiste pour l'avenir.

Il faut également savoir qu'il existe des contournements.

Il est par exemple possible de « simuler » une espace fine insécable grâce à CSS, par exemple avec un code tel que celui-ci :

```
<span style="font-size:50%;">&nbsp;</span>
```

On obtient ainsi un résultat reconnu presque universellement.

Certes le code devient plus lourd, mais cette technique pourrait par exemple être employée dans les générateurs HTML automatiques ou dans tous les cas où les fichiers HTML ne sont pas (ou très rarement) modifiés à la main.

Conclusion

Malgré quelques problèmes graves, je considère que le support de l'espace fine insécable a atteint un stade où celle-ci pourrait commencer à être utilisée sur certaines plateformes.

Je pense notamment à la plateforme Gnome et aux logiciels Mozilla.

En analysant la situation, on se rend compte qu'il ne reste « que » 3 bugs réellement bloquants (Qt, Mac OS et les TTY), que tous ont été rapportés et que l'un d'eux est d'ores et déjà corrigé en amont.

Je vous rappelle que les sources de ce document sont disponibles à cette adresse : <http://malizor.org/public/fines/>

Celles-ci (et en particulier les captures d'écran) peuvent s'avérer utiles pour documenter d'éventuels nouveaux rapports de bugs et pour aider à leur résolution.

Espérons que cet « état des lieux » permettra de faire avancer les choses et que, prochainement, les traductions de tous les logiciels libres emploieront l'espace fine insécable.

Merci pour votre attention !